CLAIMS

1. A method of embedding an image into two images, comprising:

performing a digital halftoning process on a Cartesian product of color spaces to embed the image into the two images.

2. The method of claim 1, wherein the digital halftoning process comprises a vector error diffusion method.

3. The method of claim 1, wherein the digital halftoning process comprises a modified error diffusion method.

4. The method of claim 1, wherein the digital halftoning process comprises an iterative isotropic halftoning process.

5. The method of claim 1, wherein one of said two images is a rotated version of the other of said two images.

6. The method of claim 4, wherein the iterative isotropic halftoning process comprises:

for each iteration

for each i

for each j

for each output vector $o = (o_1, o_2, o_3) \in P$

replace $Outimage_k(i, j)$ with o_k for k = 1, 2, 3,

set $Error(o) = \sum_{k=1}^{3} v_k \left\| L(Outimage_k - A_k') \right\|$

endfor

find output vector $o_{\min} = \arg\min_{o \in P} Error(o)$

set Outimage(i, j) = $o_{\min}$.

endfor (j)

endfor (i)

wherein if Outimage has not changed between two iterations or maximum number of iterations reached, then exit the iterations loop,

where:

A_1', A_2' and A_3' are input images;

P comprises a set of output vectors;

Output comprises A_1'and A_2' where (A_1,A_2,A_3) = Outimage which resembles (A_1', A_2', A_3');

v_i determines how strongly the error in each image is minimized; and

L comprises a linear space-invariant model of a human vision system.

7. The method of claim 6, wherein said Outimage is initialized using a random set of pixels.

8. The method of claim 6, wherein said Outimage is initialized using a uniform image of a single output vector.

9. The method of claim 6, wherein said Outimage is initialized by performing vector error diffusion.

10. The method of claim 6, wherein said Outimage is initialized by performing modified error diffusion.

11. The method of claim 6, wherein pixels of the input image is within a convex hull of the output vectors.

12. The method of claim 6, further comprising gamut mapping the images.

13. The method of claim 12, wherein the gamut mapping comprises:

for $p = (p_1, p_2, p_3) \in S$,

$M(p) = (s_1p_1 + d_1, s_2p_2 + d_2, s_3p_3 + d_3)$

where:

s_i comprise real numbers denoting scaling factors; and

d_i comprise offset vectors in the color space.

14. The method of claim 13, further comprising using the Qhull algorithm.

15. The method of claim 12, further comprising optimizing the gamut mapping.

16. The method of claim 15, wherein optimizing the gamut mapping comprises:

$$\max_{s_i, d_i} \min\left(\frac{s_1}{\alpha_1}, \frac{s_2}{\alpha_2}, \frac{s_3}{\alpha_3}\right) \text{ such that } M(S) \in H$$

wherein H is the convex hull of the output vectors.

17. The method of claim 15, wherein the optimizing of the gamut mapping comprises:

solving $\max_{s_i, d_i} s_1 s_2 s_3$ such that $M(S) \in H$,

wherein H comprises the convex hull of the output vectors.

18. The method of claim 1, wherein more than one image is embedded into said two images.

19. The method of claim 1, wherein said image is embedded into more than said two images.

20. The method of claim 1, wherein said images comprise color images.

21. The method of claim 1, wherein said images comprise black and white images.

22. The method of claim 1, wherein said images comprise multi-bit images.

23. A method of deploying computing infrastructure, comprising integrating computer-readable code into a computing system, wherein the code in combination with the computing system is capable of performing the method of claim 1.

24. A method of extracting an image from two images, comprising:

extracting the image from the two images using a binary operation on each pair of pixels from the two images.

25. The method of claim 24, wherein extracting the image from the two images comprises extracting the image from more than two images.

26. The method of claim 24, wherein extracting the image comprises extracting more than one image from the two images.

27. A method of embedding a color image into two color images comprising:

decomposing the color images into separate images in their color planes;

for each color plane, performing a digital halftoning process on a

Cartesian product of pixel value spaces to embed the image into the two images; and

combining the halftone images of the color planes into a single color image.

28. A method of embedding a multi-bit image into two multi-bit images, comprising:

performing a digital halftoning process on a Cartesian product of pixel value spaces to embed the image into the two images.

29. A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processor, the program comprising:

instructions for performing a digital halftoning process on a Cartesian product of color spaces to embed the image into the two images.

30. A system for embedding an image into two images comprising:

means for providing said image to be embedded into said two images; and

means for performing a digital halftoning process on the Cartesian product of color spaces to embed the image into the two images.

31. A system for embedding an image into two images comprising:

an image input device; and

a digital halftoning device that performs a digital halftoning process on a Cartesian product of color spaces to embed the image received by the image input device into the two images.

32. The system of claim 31, further comprising a gamut mapping device that performs gamut mapping on the image received by the image input device.